

09/591080
RECEIVED

JUL 31 2003

Technology Center 2100

1

DETERMINING ROUND-TRIP TIME DELAY**NOTICE OF COPYRIGHTS AND TRADE DRESS**

[0001] A portion of the disclosure of this patent document contains material which is subject to copyright protection. This patent document may show and/or describe matter which is or may become trade dress of the owner. The copyright and trade dress owner has no objection to the facsimile reproduction by any one of the patent disclosure as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright and trade dress rights whatsoever.

RELATED APPLICATION INFORMATION

[0002] This application is related to Application No. 09/591,092 filed 06/09/2000 entitled "Method of Determining Real-Time Latency and Apparatus Therefor," which is incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field Of The Invention

[0003] This invention relates, in general, to data transmission, and more particularly, to methods of determine time delay for a round-trip transmission of data and apparatuses therefor.

Description Of Related Art

[0004] A user accessing a computer server across a computer network must transmit data across the computer network from the user's computer to the computer server and must also receive data across the computer network from the computer server to the user's computer. Therefore, the user requires fast data transmission rates across the computer network and requires, in particular, fast round-trip data transmission across the computer network. However, as computer networks continuously grow in size and complexity, the data transmission rates associated with the larger and more complex computer networks may decrease. Accordingly, a need exists for a method of determining a time delay for the round-trip transmission of data and an apparatus therefor.

DESCRIPTION OF THE DRAWINGS

[0005] The invention will be better understood from a reading of the following detailed description, taken in conjunction with the accompanying drawing figures in which:

[0006] FIG. 1 illustrates a block diagram of an electronic apparatus for determining a time delay of a round-trip transmission of data in accordance with an embodiment of the invention;

[0007] FIG. 2 illustrates a flow chart for a method of determining a time delay for a round-trip transmission of data in accordance with an embodiment of the invention; and

[0008] FIGs. 3 through 6 illustrate flow charts of detailed portions of the method of FIG. 2 in accordance with an embodiment of the invention.

[0009] For simplicity and clarity of illustration, the same reference numerals in different figures denote the same elements, and descriptions and details of well-known features and techniques are omitted to avoid unnecessarily obscuring the invention.

[0010] Furthermore, the terms first, second, third, fourth, and the like in the description and in the claims, if any, are used for distinguishing between similar elements and not necessarily for describing a sequential or chronological order. However, it is understood that the embodiments of the invention described herein are capable of operation in other sequences than described or illustrated herein. It is further understood that the terms so used are interchangeable under appropriate circumstances.

DETAILED DESCRIPTION OF THE INVENTION

[0011] FIG. 1 illustrates a block diagram of an electronic device or apparatus 100 for determining a time delay of a round-trip transmission of data. Electronic apparatus 100 comprises an incoming data portion and an outgoing data portion. The incoming data portion includes a data reception portion 110, an input memory portion 115, a data validity portion 120, and, a first memory and data transfer management portion 125. Input memory portion 115 and data validity portion 120 are both coupled to data reception portion 110. Memory and data transfer management portion 125 is coupled to both of input memory portion 115 and data validity portion 120.

[0012] The outgoing data portion of electronic apparatus 100 comprises a second memory and data transfer management portion 150, an output memory portion 155, a data pattern management portion 160, a header format portion 165, and a data transmission portion 170. Memory and data transfer management portion 150 is coupled to memory and data transfer management portion 125. Output memory portion 155 is coupled to both of input memory portion 115 and memory and data transfer management portion 150. Data pattern management portion 160 is coupled to memory and data transfer management portion 150 and data transmission portion 170. Header format portion 165 is coupled to output memory portion 155, and data transmission portion 170 is coupled to header format portion 165.

[0013] In the preferred embodiment, the incoming and outgoing data portions of electronic apparatus 100 are formed within a single field-programmable gate array (FPGA),

as indicated by a dashed line 105. For example, memory portions 115, 155, memory and data transfer management portions 125, 150, data validity portion 120, data reception portion 110, data transmission portion 170, header format portion 165, and data pattern management portion 160 can be located within the single FPGA.

[0014] Electronic apparatus 100 further comprises a data pattern memory portion 190 coupling data pattern management portion 160 to data transmission portion 170. In the preferred embodiment, data pattern memory portion 190 is not included in the FPGA. Instead, data pattern memory portion 190 is preferably a separate dynamic random access memory (DRAM).

[0015] As a brief overview of the operation of electronic apparatus 100, data reception portion 110 receives an incoming data packet or frame, and data validity portion 120 validates the incoming data packet. Input memory portion 115 receives a portion of the incoming data packet from data reception portion 110, and input memory portion 115 stores the portion of the incoming data packet. The portion of the incoming data packet comprises, among other items, an Internet Protocol (IP) source address, an IP destination address, a Transport Control Protocol (TCP) source port, a TCP destination port, and a time stamp. Memory and data transfer management portions 125, 150 interact or cooperate to manage a transfer of the stored portions of the incoming data packet from input memory portion 115 to output memory portion 155. Output memory portion 155 receives the portion of the incoming data packet from input memory portion 115, and output memory portion 155 stores the portion of the incoming data packet. Header format portion 165 takes the portion of the

incoming data packet and inserts it into an outgoing data packet transmitted out of electronic apparatus 100 through data transmission portion 170. Data pattern management portion 160 manages an insertion of a data pattern from data pattern memory portion 190 into the outgoing data packet from data transmission portion 170. The operation of electronic apparatus 100 is described in more detail with reference to FIGs. 2 through 6.

[0016] FIG. 2 illustrates a flowchart for a method 200 of determining a time delay for a round-trip transmission data. A first electronic device or apparatus transmits a first data packet at a first time where the first electronic apparatus has a first IP source address and a first TCP source port. In the preferred embodiment, the first data packet comprises the first IP source address, a first IP destination address, a first IP checksum, the first TCP source port, a first TCP destination port, a first set of six TCP flags, a first TCP checksum, a first data pattern, a first time stamp indicating the first time when the first data packet was transmitted from the first electronic apparatus, and a first Checklist Redundancy Check (CRC) checksum. A second electronic device or apparatus, such as electronic apparatus 100 of FIG. 1 waits for the first data packet. The second electronic apparatus has the first IP destination address and the first TCP destination port.

[0017] At a step 205 of method 200 in FIG. 2, the second electronic apparatus begins to receive the first data packet transmitted from the first electronic apparatus. Upon beginning to receive the first data packet, the second electronic apparatus checks a status of a first memory portion within the second electronic apparatus. As an example, referring back to FIG. 1, as data reception portion 110 begins to receive the first data packet, memory and data

transfer management portion 125 checks the status of input memory portion 115. If the status of input memory portion 115 is full, then method 200 (FIG. 2) terminates or starts over by waiting for a new data packet and begins receiving the new data packet at step 205 (FIG. 2). However, if the status of input memory portion 115 is empty or if input memory portion 115 has enough empty memory to store desired portions of the first data packet, then data reception portion 110 begins identifying different portions of the first data packet while receiving the first data packet. In the preferred embodiment, input memory portion 115 is large enough to store the desired portions of two data packets. Electronic apparatus 100 stores the identified portions of the first data packet within input memory portion 115 while receiving the first data packet. Data validity portion 120 validates the different portions of the first data packet while electronic apparatus 100 receives the different portions of the first data packet.

[0018] Returning to FIG. 2, step 205 of method 200 also begins the calculation of a CRC checksum for the first data packet. This calculation begins with the first byte of data of the first data packet and preferably starts upon receiving the first byte of data of the first data packet. Next, steps 210, 215, and 220 of method 200 briefly describe the identifying, storing, and validating steps described in the previous paragraph. At step 210, the second electronic apparatus identifies, stores, and validates portions of an IP header of the first data packet, and at a step 215, the second electronic apparatus identifies, stores, and validates portions of a TCP header of the first data packet. At step 220, the second electronic apparatus identifies and stores the time stamp of the first data packet. Steps 210 and 215 are described in more detailed hereinafter with respect to FIGs. 3 and 4, respectively.

[0019] At a step 225 of method 200, the second electronic apparatus stops receiving the first data packet. Then, at a step 230, the second electronic apparatus validates the entire first data packet based on a CRC checksum match. As an example, the second electronic apparatus can perform step 230 by comparing the calculated and received CRC checksums. If the calculated and received CRC checksums are not equal to each other, then method 200 terminates or starts over by waiting for a new data packet and begins receiving the new data packet at step 205. However, if the calculated and received CRC checksums are equal to each other, then method 200 continues such that the second electronic apparatus sets or changes the status of the first memory portion storing the portions of the first data packet from empty to full.

[0020] Then, the second electronic apparatus checks a status of a second memory portion within the second electronic apparatus. If the status of the second memory portion is full, then the second electronic apparatus waits until at least a portion of the second memory portion is free, is empty, or otherwise becomes available. This portion of the second memory portion needs to be large enough to store the portions of the first data packet currently stored in the first memory portion. After the portion of the second memory becomes available, the second electronic apparatus transfers the stored portions of the first data packet from the first memory portion to the second memory portion. Then, the second electronic apparatus sets or changes the status of the second memory portion from empty to full, and the second electronic apparatus also sets or changes the status of the first memory portion from full to empty. As an example, referring back to FIG. 1, memory and data transfer management portions 125, 150 cooperate or interact to transfer the stored portions of the first data packet

from input memory portion 115 to output memory portion 155. In the preferred embodiment, steps 210, 215, and 220 in FIG. 2 are performed in real-time while simultaneously receiving the first data packet.

[0021] Returning to FIG. 2, method 200 continues at a step 235 where the second electronic apparatus begins transmitting a second data packet back to the first electronic apparatus. Step 235 of method 200 also begins the calculation of a CRC checksum for the second data packet. This calculation begins with the first byte of data of the second data packet. At a step 240, the second electronic apparatus inserts the stored portions of the IP header of the first data packet into an IP header of the second data packet, and at a step 245, the second electronic apparatus inserts the stored portions of the TCP header of the first data packet into a TCP header of the second data packet. As an example, header format portion 165 (FIG. 1), output memory portion 155 (FIG. 1), and data transmission portion 170 (FIG. 1) can perform steps 240 and 245 in FIG. 2. Steps 240 and 245 are described in more detailed hereinafter with respect to FIGs. 5 and 6, respectively.

[0022] Returning back to FIG. 2, at a step 250 of method 200, the second electronic apparatus sends or transmits a second data pattern as part of the second data packet. The second data pattern of the second data packet can be the same as or different from the first data pattern in the first data packet. As an example, data pattern management portion 160 (FIG. 1), data pattern memory portion 190 (FIG. 1), and data transmission portion 170 (FIG. 1) can perform step 250.

[0023] Subsequently, at a step 255 of method 200, the second electronic apparatus inserts the first time stamp of the first data packet stored in the second memory portion as a second time stamp in the second data packet. Next, at a step 260, the second electronic apparatus inserts a validity check for the second data packet into the second data packet. As an example, the validity check is a second CRC checksum that is different from the first CRC checksum of the first data packet. In the preferred embodiment, header format portion 165 (FIG. 1), output memory portion 155 (FIG. 1), and data transmission portion 170 (FIG. 1) perform steps 255 and 260. Subsequently, at a step 265 of method 200, the second electronic apparatus stops transmitting the second data packet. In the preferred embodiment, steps 240, 245, 250, 255, and 260 are performed in real-time while simultaneously transmitting the second data packet.

[0024] Next, the first electronic apparatus receives the second data packet at a second time. This second time occurs after the first time at which the first electronic apparatus originally transmitted the first data packet to the second electronic apparatus. The first electronic apparatus determines the time delay for the round-trip transmission of data from the first electronic apparatus to the second electronic apparatus and back to the first electronic apparatus by subtracting the time indicated by the second time stamp in the second data packet from the second time. As indicated earlier at step 255, the second time stamp in the second data packet contains the first time at which the first electronic apparatus transmitted the first data packet.

[0025] FIG. 3 illustrates a flowchart of substeps within step 210 of FIG. 2. At a step 310 in FIG. 3, the second electronic apparatus identifies a beginning of the IP header in the first data packet, and at a step 320, the second electronic apparatus begins calculating an IP checksum for the first data packet. At a step 330, the second electronic apparatus identifies an IP source address within the IP header of the first data packet, and at a step 340, the second electronic apparatus stores the first IP source address. Then, at a step 350, the second electronic apparatus identifies an IP destination address in the IP header of the first data packet, and at a step 360, the second electronic apparatus stores the IP destination address. Next, at a step 370, the second electronic apparatus identifies an end of the IP header, and, at a step 380, the second electronic apparatus validates the IP header data of the first data packet based on an IP checksum match.

[0026] To perform step 380, the second electronic apparatus finishes calculating the IP checksum for the first data packet and compares the calculated IP checksum to the received IP checksum of the first data packet. If the calculated and received IP checksums are equal to each other, then the IP checksum of the first data packet is valid, and method 200 (FIG. 2) continues with step 215 (FIG. 2). However, if the calculated and received IP checksums are not equal to each other, then method 200 (FIG. 2) terminates or starts over, and the second electronic apparatus waits for another data packet and begins receiving the other data packet at step 205 (FIG. 2). In the preferred embodiment, steps 310, 320, 330, 340, 350, 360, 370, and 380 are performed in real-time while simultaneously receiving the first data packet. Also in the preferred embodiment, the second electronic apparatus identifies and stores the IP source and destination addresses before validating the IP header data.

[0027] FIG. 4 illustrates a flowchart of the substeps within step 215 of FIG. 2. At a step 410 of FIG. 4, the second electronic apparatus identifies a beginning of the TCP header in the first data packet. Then, at a step 415, the second electronic apparatus begins to calculate a TCP checksum for the first data packet. Next, at a step 420, the second electronic apparatus identifies a TCP source port in the TCP header of the first data packet, and at a step 430, the second electronic apparatus stores the TCP source port. At a step 440, the second electronic apparatus identifies a TCP destination port in the TCP header of the first data packet, and at a step 450, the second electronic apparatus stores the TCP destination port. Then, at a step 460, the second electronic apparatus identifies the TCP flags in the TCP header of the first data packet, and at a step 470, the second electronic apparatus stores at least a portion of the TCP flags. In the preferred embodiment, the second electronic apparatus receives six TCP flags in the first data packet, but stores only two of the six TCP flags. In particular, the second electronic apparatus stores the TCP flags identified as a final (FIN) flag and a synchronous (SYN) flag. Next, at a step 480, the second electronic apparatus identifies an end of the TCP data, and at a step 490, the second electronic apparatus validates the TCP data, including the TCP header, in the first data packet based on a TCP checksum match.

[0028] As an example, the second electronic apparatus can perform step 490 by comparing the calculated and received TCP checksums. If the calculated and received TCP checksums are equal to each other, then method 200 (FIG. 2) continues with step 220 (FIG. 2). However, if the calculated and received checksums are not equal to each other, then method 200 (FIG. 2) terminates or starts over, and the second electronic apparatus waits to receive another data packet and begins receiving the new data packet at step 205 (FIG. 2).

Also in the preferred embodiment, the second electronic apparatus performs steps 410, 420, 430, 440, 450, 460, 470, 480, and 490 in real-time while simultaneously receiving the first data packet. Furthermore, the second electronic apparatus preferably identifies and stores the TCP source and destination ports and the TCP flags before validating the TCP data.

[0029] FIG. 5 illustrates a flowchart of the substeps in step 240 of FIG. 2. At a step 510 of FIG. 6, the second electronic apparatus counts an IP header offset, and at a step 520, the second electronic apparatus calculates a second IP checksum for the second data packet. Step 520 can be performed at this time because the portions of the IP header used to calculate IP checksum are already known and stored in the second memory portion. Next, at a step 530, the second electronic apparatus adds an IP checksum offset to the IP header offset, and at a step 540, the second electronic apparatus inserts the calculated IP checksum into the second data packet. Next, at a step 550, the second electronic apparatus adds an IP source address offset to the previous offset sum, and then the second electronic apparatus uses the first IP destination address of the first data packet stored in the second memory portion. In particular, at a step 560, the second electronic apparatus inserts the first IP destination address as a second IP source address in the second data packet.

[0030] Then, at a step 570, the second electronic apparatus adds an IP destination address offset to the previous offset sum, and then the second electronic apparatus uses the first IP source address of the first data packet stored in the second memory portion. In particular, at a step 580, the second electronic apparatus inserts the first IP source address as a second IP destination address in the second data packet. In the preferred embodiment, the second

electronic apparatus performs steps 520, 530, and 540 before steps 550, 560, 570, and 580. Also in the preferred embodiment, the second electronic apparatus performs steps 510, 520, 530, 540, 550, 560, 570, and 580 in real-time while simultaneously transmitting the second data packet.

[0031] FIG. 6 illustrates a flowchart of the substeps of step 245 in FIG. 2. At a step 610 in FIG. 6, the second electronic apparatus counts a TCP header offset, and at a step 620, the second electronic apparatus adds a TCP source port offset to the TCP header offset. Then, the second electronic apparatus uses the first TCP destination port of the first data packet stored in the second memory portion. In particular, at a step 630, the second electronic apparatus inserts the first TCP destination port as a second TCP source port in the second data packet. Next, at a step 640, the second electronic apparatus adds a TCP destination port offset to the previous offset sum, and then the second electronic apparatus uses the first TCP source port of the first data packet stored in the second memory portion. In particular, at a step 650, the second electronic apparatus inserts the first TCP source port as a second TCP destination port in the second data packet.

[0032] Subsequently, at a step 660, the second electronic apparatus adds a TCP flag offset to the previous offset sum, and then the second electronic apparatus uses the two TCP flags of the first data packet stored in the second memory portion. In particular, at a step 670, the second electronic apparatus inserts the FIN flag and the SYN flag as a portion of the second TCP flags into the second data packet. The second electronic apparatus also inserts four other TCP flags, for a total of six TCP flags, into the second data packet. In particular,

the second electronic apparatus inserts a TCP flag identified as an acknowledgment (ACK) flag where the ACK flag has a value of one. The second electronic apparatus also inserts three other TCP flags, each having a value of zero.

[0033] Then, at a step 680, the second electronic apparatus adds a TCP checksum offset to the previous offset sum, and at a step 690, the second electronic apparatus calculates and inserts the second TCP checksum into the second data packet. In the preferred embodiment, the second electronic apparatus component begins and finishes calculating the second TCP checksum after step 680. Also in the preferred embodiment, the second electronic apparatus performs steps 610, 620, 630, 640, 650, 660, and 670 before steps 680 and 690. Furthermore, the second electronic apparatus preferably performs steps 610, 620, 630, 640, 650, 660, 670, 680, and 690 in real-time while simultaneously transmitting the second data packet.

[0034] Therefore, an improved method of determining a time delay for the round-trip transmission of data and an apparatus therefor are provided to overcome the disadvantages of the prior art. The method and apparatus enable the detection of an increase or decrease in the time delay for the round-trip transmission of data across a computer network.

[0035] Although the invention has been described with reference to specific embodiments, it will be understood by those skilled in the art that various changes may be made without departing from the spirit or scope of the invention. For instance, the numerous details set forth herein such as, for example, the specific sequence of steps are provided to facilitate the understanding of the invention and are not provided to limit the scope of the invention. Furthermore, the method described herein is not limited to the round-trip

transmission of data between two electronic devices. Instead, the method can be modified and applied to the round-trip or non-round-trip transmission of data between three or more electronic devices. Accordingly, the disclosure of embodiments of the invention is intended to be illustrative of the scope of the invention and is not intended to be limiting. It is intended that the scope of the invention shall be limited only to the extent required by the appended claims.

CLAIMS

It is claimed:

1. A method of determining a time delay for a round-trip transmission of data comprising:

receiving a first data packet comprising a first IP source address, a first IP destination address, a first TCP source port, a first TCP destination port, and a first time stamp indicating a first time when the first data packet was transmitted;

inserting the first IP destination address as a second IP source address in a second data packet;

inserting the first IP source address as a second IP destination address in the second data packet;

inserting the first TCP destination port as a second TCP source port in the second data packet;

inserting the first TCP source port as a second TCP destination port in the second data packet;

inserting the first time stamp as a second time stamp in the second data packet; and

transmitting the second data packet.

2. The method of 1 further comprising:

transmitting the first data packet at the first time;

receiving the second data packet at a second time; and

determining a difference between the first time in the second time stamp and the second time to establish the time delay for the round-trip transmission of data.

3. The method of 1 further comprising:

validating the first IP destination address while receiving the first data packet, before inserting the first IP destination address, before inserting the first IP source address, before inserting the first TCP destination port, before inserting the first TCP source port, and before transmitting the second data packet; and

validating the first TCP destination port while receiving the first data packet, before inserting the first IP destination address, before inserting the first IP source address, before inserting the first TCP destination port, before inserting the first TCP source port, and before transmitting the second data packet.

4. The method of 1 wherein:

inserting the first IP destination address occurs while transmitting the second data packet; and

inserting the first IP source address occurs while transmitting the second data packet.

5. The method of 1 wherein:

inserting the first TCP destination port occurs while transmitting the second data packet; and

inserting the first TCP source port occurs while transmitting the second data packet.

6. The method of 1 wherein:

inserting the first time stamp occurs while transmitting the second data packet.

7. The method of 1 further comprising:

providing the first data packet further comprise a first IP checksum, a first TCP checksum, and a first CRC checksum;

validating the first IP checksum while receiving the first data packet;

validating the first TCP checksum while receiving the first data packet; and

validating the first CRC checksum.

8. The method of 7 further comprising:

storing the first IP source address and the first IP destination address before validating the first IP checksum; and

storing the first TCP source port and the first TCP destination port after validating the first IP checksum and before validating the first TCP checksum.

9. The method of 7 wherein:

validating the first TCP checksum occurs after validating the first IP checksum and before validating the first CRC checksum.

10. The method of 7 wherein:

validating the first CRC checksum occurs after receiving the first data packet.

11. The method of 7 further comprising:

determining a second IP checksum for the second data packet;

inserting the second IP checksum into the second data packet while transmitting the second data packet;

determining a second TCP checksum for the second data packet; and

inserting the second TCP checksum into the second data packet while transmitting the second data packet.

12. The method of 11 wherein:

inserting the first IP destination address occurs while transmitting the second data packet;

inserting the first IP source address occurs while transmitting the second data packet;

inserting the first TCP destination port occurs while transmitting the second data packet;

inserting the first TCP source port occurs while transmitting the second data packet; and

inserting the first time stamp occurs while transmitting the second data packet.

13. The method of 11 wherein:

inserting the first IP destination address occurs after inserting the second IP checksum;

inserting the first IP source address occurs after inserting the first IP destination address;

inserting the first TCP destination port occurs after inserting the first IP source address; and

inserting the first TCP source port occurs after inserting the first TCP destination port and before inserting the second TCP checksum.

14. The method of 1 further comprising:

providing the first data packet to further comprise a first data pattern; and
inserting a second data pattern into the second data packet.

15. The method of 14 wherein:

inserting the second data pattern occurs while transmitting the second data packet.

16. The method of 1 further comprising:

providing the first data packet to further comprise a first TCP flag; and
inserting the first TCP flag as a second TCP flag into the second data packet.

17. The method of 16 further comprising:

validating the first TCP flag while receiving the first data packet, before inserting the first IP destination address, before inserting the first IP source address, before inserting the first TCP destination port, before inserting the first TCP source port, before transmitting the second data packet, and before inserting the first TCP flag.

18. The method of 16 wherein:

inserting the second TCP flag occurs while transmitting the second data packet.

19. The method of 1 further comprising:

providing the first data packet to further comprise two TCP flags;
inserting the two TCP flags into the second data packet;
inserting an additional TCP flag into the second data packet, the additional TCP flag
having a value of one; and

inserting three additional TCP flags into the second data packet, the three additional TCP flags each having a value of zero.

20. The method of 1 further comprising:

providing the first data packet to further comprising six TCP flags;

inserting two of the six TCP flags into the second data packet;

inserting an additional TCP flag into the second data packet, the additional TCP flag having a value of one; and

inserting three additional TCP flags into the second data packet, the three additional TCP flags each having a value of zero.

21. The method of 20 further comprising:

providing a FIN flag and a SYN flag for the two of the six TCP flags; and

providing an ACK flag for the additional TCP flag.

22. The method of 1 further comprising:

providing the first data packet to further comprise a first IP checksum, first TCP flags, a first TCP checksum, and a first CRC checksum;

validating the first IP checksum while receiving the first data packet;

validating the first TCP checksum while receiving the first data packet;

validating the first CRC checksum after receiving the first data packet;

determining a second IP checksum for the second data packet;

inserting and the second IP checksum into the second data packet while transmitting the second data packet;

inserting the first TCP flags as second TCP flags into the second data packet while transmitting the second data packet;

determining a second TCP checksum for the second data packet;

inserting the second TCP checksum into the second data packet while transmitting the second data packet;

determining a second CRC checksum for the second data packet; and

inserting the second CRC checksum into the second data packet while transmitting the second data packet.

23. The method of 22 wherein:

inserting the first IP destination address occurs while transmitting the second data packet;

inserting the first IP source address occurs while transmitting the second data packet;

inserting the first TCP destination port occurs while transmitting the second data packet;

inserting the first TCP source port occurs while transmitting the second data packet;

and

inserting the first time stamp occurs while transmitting the second data packet.

24. The method of 23 wherein:

inserting the first IP destination address occurs after inserting the second IP checksum;

inserting the first IP source address occurs after inserting the first IP destination address;

inserting the first TCP destination port occurs after inserting the first IP source address;

inserting the first TCP source port occurs after inserting the first TCP destination port;

inserting the first TCP flags occurs after inserting the first TCP source port;

inserting the second TCP checksum occurs after inserting the first TCP flags;

inserting the first time stamp occur after inserting the second TCP checksum; and

inserting the second CRC checksum occurs after inserting the first time stamp.

25. The method of 24 further comprising:

providing the first data packet to further comprise a first data pattern; and

inserting a second data pattern into the second data packet while transmitting the second data packet.

26. The method of 25 further comprising:

transmitting the first data packet at the first time from a first electronic apparatus having the first IP source address and the first TCP source port;

receiving the second data packet at a second time and at the first electronic apparatus having the second IP destination address and the second TCP destination port; and

subtracting the first time in the second time stamp from the second time to determine the time delay for the round-trip transmission of data,

wherein:

receiving the first data packet further comprises receiving the first data packet at a second electronic apparatus having the first IP destination address and the first TCP destination port; and

transmitting the second data packet further comprises transmitting the second data packet from the second electronic apparatus having the second IP source address and the second TCP source port.

27. The method of 22 wherein:

inserting the first IP destination address occurs after inserting the second IP checksum;

inserting the first IP source address occurs after inserting the first IP destination address;

inserting the first TCP destination port occurs after inserting the first IP source address;

inserting the first TCP source port occurs after inserting the first TCP destination port;

inserting the first TCP flags occurs after inserting the first TCP source port;

inserting the second TCP checksum occurs after inserting the first TCP flags;

inserting the first time stamp occur after inserting the second TCP checksum; and

inserting the second CRC checksum occurs after inserting the first time stamp.

28. The method of 22 further comprising:

providing the first data packet to further comprise a first data pattern; and

inserting a second data pattern into the second data packet while transmitting the second data packet.

29. The method of 22 further comprising:

transmitting the first data packet at the first time from a first electronic apparatus having the first IP source address and the first TCP source port;

receiving the second data packet at a second time and at the first electronic apparatus having the second IP destination address and the second TCP destination port; and

subtracting the first time in the second time stamp from the second time to determine the time delay for the round-trip transmission of data,

wherein:

receiving the first data packet further comprises receiving the first data packet at a second electronic apparatus having the first IP destination address and the first TCP destination port; and

transmitting the second data packet further comprises transmitting the second data packet from the second electronic apparatus having the second IP source address and the second TCP source port.

30. The method of 1 further comprising:

waiting for the first data packet;

checking a status of a first memory portion;

storing a portion of the first data packet if the first memory portion is available, the portion of the first memory portion comprising the first IP source address, the first IP destination address, the first TCP source port, and the first TCP destination port;

checking a validity of the first data packet;

setting the status of the first memory portion to full if the first data packet is valid;

checking a status of a second memory portion;

transferring the portion of the first data packet from the first memory portion to the second memory portion if the second memory portion is available and if the first data packet is valid;

setting the status of the second memory portion to full; and

setting the status of the first memory portion to empty.

31. An electronic apparatus for determining a time delay for a round-trip transmission of data comprising:

an output memory portion for receiving a portion of an incoming data packet

a data pattern management portion for managing an insertion of a data pattern into an outgoing data packet by

selecting a first source value, a first destination value and a first time stamp from the incoming data packet

setting a second source value to be the first destination value

setting a second destination value to be the first source value

setting a second time stamp to be the first time stamp

a header format portion for inserting the second source value, the second destination value and the second time stamp into the outgoing data packet.

32. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 further comprising:

an incoming data portion comprising:

a data reception portion for receiving the incoming data packet;

an input memory portion for storing a portion of an incoming data packet;

a data validity portion for validating the incoming data packet;

an outgoing data portion comprising:

the output memory portion;

the data pattern management portion;

the header format portion;

a data transmission portion for transmitting the outgoing data packet.

33. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 wherein:

the first source value comprises a TCP source port for the incoming data packet

the first destination value comprises a TCP destination port for the incoming data packet

the second source value comprises a TCP source port for the outgoing data packet

the second destination value comprises a TCP destination port for the outgoing data packet.

34. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 wherein:

the input memory portion, the output memory portion, the header format portion, and the data pattern management portion are located within a field-programmable gate array.

35. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 wherein

the data pattern management portion is further for

selecting a first set of TCP flags from the incoming data packet

setting a second set of TCP flags to be the first set of TCP flags

the header format portion is further for inserting the second set of TCP flags into the outgoing data packet.

36. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 wherein:

the first source value comprises an IP source address for the incoming data packet

the first destination value comprises an IP destination address for the incoming data packet

the second source value comprises an IP source address for the outgoing data packet

the second destination value comprises an IP destination address for the outgoing data packet.

37. The electronic apparatus for determining a time delay for a round-trip transmission of data of 31 wherein:

38. A method of determining a time delay for a round-trip transmission of data comprising:

receiving a first data packet, the first data packet comprising

a first source value

a first destination value

a first time stamp indicating a first time when the first data packet was

transmitted

preparing a second data packet, the second data packet comprising

a second source value

a second destination value

a second time stamp

setting the first destination value as the second source value in the second data packet

setting the first source value as the second destination value in the second data packet

setting the first time stamp as the second time stamp in the second data packet

transmitting the second data packet.

39. The method of determining a time delay for a round-trip transmission of data of 38 further comprising validating the first destination value before inserting the first destination value, before inserting the first source value, and before transmitting the second data packet

40. The method of determining a time delay for a round-trip transmission of data of 38 further comprising:

transmitting the first data packet at a first time
receiving the second data packet at a second time
determining a difference between the first time in the second time stamp and the
second time to establish the time delay for the round-trip transmission of data.

41. The method of determining a time delay for a round-trip transmission of data of 38
wherein

inserting the first destination value occurs while transmitting the second data packet
inserting the first source value occurs while transmitting the second data packet.

42. The method of determining a time delay for a round-trip transmission of data of 38
wherein inserting the first time stamp occurs while transmitting the second data packet.

43. The method of determining a time delay for a round-trip transmission of data of 38
further comprising

providing the first data packet to further comprise a first data pattern
inserting a second data pattern into the second data packet.

44. The method of determining a time delay for a round-trip transmission of data of 43
wherein inserting the second data pattern occurs while transmitting the second data packet.

45. The method of determining a time delay for a round-trip transmission of data of 38,
wherein in the providing step, the first data packet further comprises a first TCP flag, the
method further comprising inserting the first TCP flag as a second TCP flag into the second
data packet.

46. An apparatus for determining a time delay for a round-trip transmission of data comprising:

means for receiving a first data packet, the first data packet comprising

a first source value

a first destination value

a first time stamp indicating a first time when the first data packet was

transmitted

means for preparing a second data packet comprising a second source value, a second destination value, a second time stamp

means for inserting the first destination value as the second source value in the second data packet

means for inserting the first source value as the second destination value in the second data packet

means for inserting the first time stamp as the second time stamp in the second data packet

means for transmitting the second data packet.

47. The apparatus for determining a time delay for a round-trip transmission of data of 46 further comprising:

means for transmitting the first data packet at a first time

means for receiving the second data packet at a second time .

means for determining a difference between the first time in the second time stamp and the second time to establish the time delay for the round-trip transmission of data.

48. The apparatus for determining a time delay for a round-trip transmission of data of 46 further comprising means for validating the first destination value while receiving the first data packet, before inserting the first destination value, before inserting the first source value, and before transmitting the second data packet

49. The apparatus for determining a time delay for a round-trip transmission of data of 46 wherein

the means for inserting the first destination value operates concurrently with the means for transmitting the second data packet

the means for inserting the first source value operates concurrently with the means for transmitting the second data packet.

50. The apparatus for determining a time delay for a round-trip transmission of data of 46 wherein the means for inserting the first time stamp operates concurrently with the means for transmitting the second data packet.

51. The apparatus for determining a time delay for a round-trip transmission of data of 46, wherein the first data packet further comprises a first data pattern, the method further comprising means for inserting a second data pattern into the second data packet.

52. The apparatus for determining a time delay for a round-trip transmission of data of 51 wherein the means for inserting the second data pattern operates concurrently with the means for transmitting the second data packet.

53. The apparatus for determining a time delay for a round-trip transmission of data of 46, wherein the first data packet further comprises a first TCP flag, the apparatus further comprising means for inserting the first TCP flag as a second TCP flag into the second data packet.

54. The apparatus for determining a time delay for a round-trip transmission of data of 46 comprising a field programmable gate array.